

```

# Simulates the 15 day mouse exposure study
# Data collected during and after exposure on 1st day
# and at end of exposure on day 5 and 15 (1 day nose-only)
#Uses Table 3 metabolism rates

source("initialize.R") # script that loads dll, etc.

#Scenario specific values
tstart <- 0.0
tstop <- 5.95
times <- seq(tstart, tstop , by=0.05)
nend=length(times)

#load the parameters
source('./params/Mouse.R') # Physiological parameters
source('./states.R')

# timing variables for forcing functions
dstart <- tstart
dlength <- 6.0      #hours per day to expose
ddaysperwk <- 5    #days of week to expose
dexpend <- 19      #days of exposure
parms["TSTOP"] <- tstop

# Source forcing functions
# this loads the function forcing() in the namespace
source("forfunc.R")

source('./params/Fmouse_parms.R') # script with default female mouse metabolic
parameters

#Scenario Specific Parameters
parms["BW"]<- 0.022      #measured in the study
parms["QPC"]<- 37.6       #measured in the study
parms["QCC"]<- 25.9       #V/Q Ratio Marino et al. 2006

#Scenario Specific Exposure

cs=c(12.3, 32.0, 90.0) # exposure concentrations (ppm)

pname <- c("BW", "QPC", "QCC", "QLC", "QFC", "QSC", "QKC",
          "VLC", "VLUC", "VFC", "VRC", "VSC", "VKC",
          "PL", "PLU", "PF", "PS", "PR", "PB", "PK",
          "VMAXC", "KM", "VMAXCLU", "KMLU", "KFKIC" )

pval <- parms[pname] # vector with baseline values of params in pname
pout <- array(0,c(length(pname)+1,3)) # empty results array, +1 row for base values

```

```

colnames(pout)<- c("12.3 ppm", "32.0 ppm","90.0 ppm")
rownames(pout)<- c("base",pname) # first row is base values, then names in pname

for(i in 1:3){
  # Run base model
  parms["CONC"]<- cs[i]

  out <- ode(Y, times, func = "derivs", parms = parms, method="vode", atol=1.0e-10,
  rtol=1.0e-8,
  dllname = mName, initforc="initforc", forcings=forcings,
  initfunc = "initmod", nout = length(Outputs),
  outnames = Outputs)

  pout["base",i] <- out[nend, "CV"] # first row is base values, *CV* is variable
  analyzed

  # run model with each of pval varied, then re-set
  for(pn in pname){
    parms[pn] <- pval[pn]*1.01

    out <- ode(Y, times, func = "derivs", parms = parms, method="vode", atol=1.0e-10,
    rtol=1.0e-8,
    dllname = mName, initforc="initforc", forcings=forcings,
    initfunc = "initmod", nout = length(Outputs),
    outnames = Outputs)

    pout[pn,i] <- out[nend, "CV"]
    parms[pn] <- pval[pn] # reset param to base value
  }
}

pout <- pout*1000/parms["MW"] # convert units to uM

#unload the model dll
dyn.unload(paste0(mName,.Platform$dynlib.ext))

write.csv(pout,file='mouseinvivo_sa.csv')

```